

Claims

What is claimed is:

1. A method of managing a locked resource in a distributed environment, the method comprising:

receiving a request to access the resource, wherein the request originates from a requesting client computer system;

5 determining whether the resource has a conflicting lock;

if the resource has a conflicting lock, returning lock information to the requesting client computer system, so that the retry strategy of the requesting client computer system may be modified;

if the resource does not have a conflicting lock, performing the requested access.

2. A method as defined in claim 1 wherein the lock information is related to expected lifetime of the lock.

3. A method as defined in claim 2 wherein the lock owner sets the expected lifetime of the lock.

4. A method as defined in claim 2 wherein the requesting client computer system modifies a request strategy based on the returned information.

5. A method as defined in claim 4 wherein the search strategy relates to the time period between requests for the resource.

6. A method as defined in claim 1 wherein the lock property relates to the sharing property values of the lock.

7. A method as defined in claim 6 wherein the search strategy relates to the type of request.

8. A computer program product readable by a computer and encoding instructions for executing the method recited in claim 1.

9. A computer program product readable by a computer and encoding instructions for executing the method recited in claim 5.

10. A computer-readable medium having stored thereon a locked resource, wherein the locked resource comprises:

a resource object data section for storing actual object data; and

a lock object, wherein the lock object may comprise an expected lifetime property.

11. A method of allocating access to a resource in a distributed environment, the method comprising:

receiving a request to access the resource, wherein the request originates from a requesting client computer system;

determining whether the resource has a conflicting lock;

if the resource has a conflicting lock, blocking the resource for the requesting client computer system until the resource is free; and

performing the requested access, allocating a new lock to the requesting computer system.

12. A method as defined in claim 11 wherein the request for access to the resource further comprises a request to block the resource.

13. A method as defined in claim 12 wherein the request to block the resource is a predetermined header having a time value for defining a time period to block the resource.

14. A method of unlocking a locked resource in a distributed environment, the locked resource having a lock object associated with a lock owner, the method comprising:

receiving a request to access the locked resource, wherein the request originates from a requesting client computer system other than the lock owner and wherein the

5 request comprises a request to break the lock object;

identifying the request to break the lock object;

determining whether the requesting client computer system is cleared to break the lock object; and

10 removing the lock object from the resource if the requesting client computer system is cleared to break the lock object.

15. A method as defined in claim 14 further comprising:

notifying the lock owner that the lock object of the request to break the lock before removing the lock object.

16. A method as defined in claim 15 wherein the lock object is not removed for a predetermined time following notifying the lock owner of the request to break the lock.

17. A method as defined in claim 15 wherein the lock object has a timeout property value and the timeout property value is modified to effectively remove the lock object.

18. A computer program product readable by a computer and encoding instructions for executing the method recited in claim 14.

19. A system for managing resources in a distributed environment, the distributed environment having a plurality of resources and wherein at least one resource is associated with a lock object, the system comprising:

a receive module for receiving a request from a requesting client application

5 program to access at least one resource in the distributed environment;

a determination module for determining whether the resource has a conflicting
lock object associated with the requested resource; and

a communication module for returning lock information to the client application
program if the resource has a conflicting lock.

20. A system as defined in claim 19 wherein:

an owning client application program owns a lock object for the requested
resource;

5 the lock information returned to the requesting client application program relates
to the expected lifetime of the lock; and

the owning client application program determines the expected lifetime of the
lock object.

21. A system as defined in claim 20 wherein the requesting client application program
modifies a request strategy based on received information from the communication
module.

22. A system as defined in claim 19 further comprising:

a blocking module for blocking the locked resource for the requesting client
application program until the resource is released; and

5 an allocation module for allocating a new lock to the requesting client application
program following the release of the resource.

23. A system as defined in claim 22 wherein the requesting client application program
requests to block the resource when requesting access to the resource.

24. A system as defined in claim 19 wherein the receive module is adapted to receive a request to break an existing lock object; the system further comprising:

a breaking module for removing the existing lock object for the requested resource in response to a received request to break an existing lock object.

25. A system as defined in claim 24 wherein the requesting client application program requests to break the existing lock object associated with the requested resource.

26. A system as defined in claim 25 further comprising a determination module that determines whether the requesting client application program is suitably authorized and

wherein the existing lock object is not removed in response to the request to break the lock object unless the requesting client application program is suitably authorized.